PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G09G 5/00 | A1

(11) International Publication Number: **WO 99/39328**

(43) International Publication Date: 5 August 1999 (05.08.99)

(21) International Application Number: PCT/US99/01848

(22) International Filing Date: 29 January 1999 (29.01.99)

(30) Priority Data:

60/073,289	31 January 1998 (31.01.98)	US
09/240,480	29 January 1999 (29.01.99)	US

(71)(72) Applicant and Inventor: LINDSAY, Don, William [US/US]; 667 26th Street, Ogden, UT 84401 (US).

(74) Agents: MASCHOFF, Eric, L. et al.; Workman, Nydegger & Seeley, 1000 Eagle Gate Tower, 60 East South Temple, Salt Lake City, UT 84111 (US).

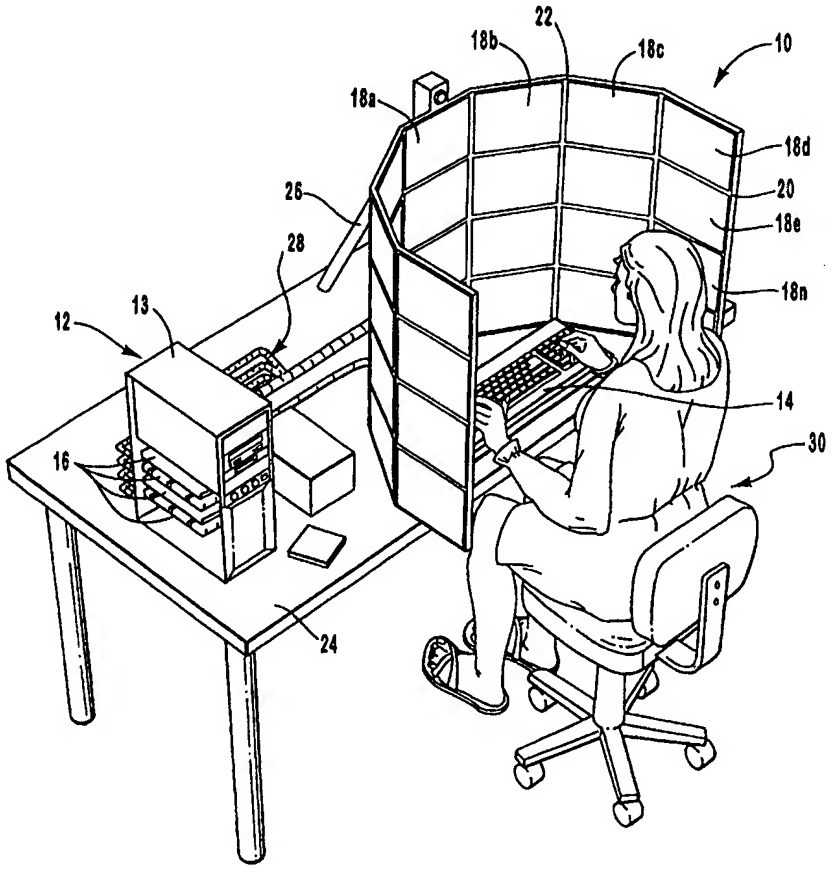
(81) Designated States: AU, CA, JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: MULTISCREEN DISPLAY SYSTEM AND METHOD

(57) Abstract

A multi-screen display system (10) comprises a plurality of display screens (18a–n) interconnected to form one large screen, preferred a concave shape about the user (30). Stored within memory of a computer (12) is each screen's position data relative to the assumed user position. Optionally, an adjustment data set can be created and used to update the screen position data so that it designates the actual user position instead of the assumed user position. Each screen is connected with the computer's graphics control circuitry (16) which comprises a graphics processor, either two-dimensional (44a–n), three-dimensional (46a–n) or both, that is capable of causing image data to be displayed on a corresponding screen. The graphics control circuitry further comprises a multi-screen controller circuit (52) that receives image data from an application program executing on the computer and then forwards that image data to the correct graphics processor for display. In this way, the system can display a single contiguous image or many two-dimensional or three-dimensional images simultaneously.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

MULTISCREEN DISPLAY SYSTEM AND METHOD

BACKGROUND OF THE INVENTION

1. The Field of the Invention

The present invention is directed to video displays for use in connection with desktop or personal computers. More particularly, the present invention is directed to a display system and method in which multiple individual video displays are interconnected so as to together function as a single contiguous video display screen that can be oriented about the computer user for displaying two-dimensional and/or complex three dimensional images.

2. The Prior State of the Art

As is well known, desktop and portable personal computers are used in connection with a wide variety of business, home, and entertainment software applications. To one degree or another, software applications require the use of a computer video display screen to display information to the computer user, typically in the form of various combinations of data, text and graphics images. And, as the sophistication and number of applications available continues to increase, the need to display greater amounts of data and/or graphics images also expands. For instance, spreadsheet, database, computer aided design, Internet browser and related applications often are best used with large video display screens to present relevant and contiguous amounts of data to the user. More often than not, however, the screen size of currently available video displays limits the user's ability to display an entire wordprocessing document, or spreadsheet, for instance. Instead, the user must continuously scroll through an application to display relevant portions. This is time consuming and reduces the computer user's productivity.

The size of currently available video displays are limiting in other respects as well. Today's windows-based graphics user interfaces and multitasking operating systems allow a user to simultaneously execute a number of different programs within multiple corresponding application "windows." For instance, a user may be simultaneously executing an Internet browser, an electronic mail application, a spreadsheet application, and a word processing application within different application windows. With such an environment, it would be advantageous for a user to keep certain application windows visible at all times, even while working in another application window. Again, even with the largest display screens currently available, the user is typically unable to display more than one or two application windows at any given time. As such, the user is forced to continuously toggle between application windows to view

and/or interact with the various applications. This can greatly reduce a computer user's efficiency and productivity and detracts from many of the advantages otherwise provided by today's desktop computers and multitasking operating systems.

In addition to the inefficiencies associated with limitations in display screen size, there are other drawbacks associated with currently available computer displays. For instance, sophisticated video entertainment and gaming applications, and related high-end three-dimensional and so-called virtual reality-type software applications, are particularly dependent on having a suitable display environment -- typically not provided with a single, flat video display screen. To address the inadequacies of currently available video displays, such applications may be written for use with a "head mounted display" (HMD), which provides a user with peripheral vision of graphics scenes, and a more realistic, "immersive" visual experience. However, such displays are not entirely satisfactory for a number of reasons, and are thus not an attractive alternative for some users. First, HMDs are effectively and practically limited to being used with specific types of applications; they are not practical for day-to-day use of in connection with standard 2D-type office applications, such as word processors. Second, even short term use of HMDs can be uncomfortable, and can cause eye strain, headaches and even nausea in some users.

In addition, use of a HMD device to view a complex three-dimensional application does not permit the user to simultaneously display another two-dimensional application. For instance, while executing a three-dimensional virtual reality application, it may be advantageous for a user to simultaneously execute an unrelated, two-dimensional windowed application, such as an electronic mail package, that the user can monitor while executing the 3D application. This capability is typically not available with a HMD, or other currently available flat video display screens that are executing complex 3D software applications.

Thus, it would be an improvement in the art to provide a computer video display and operating method that provides a display screen that is large enough to show large, contiguous views of executing application programs. In this way, a user may view an entire spreadsheet's contents, or simultaneously view a number of document pages in a side-by-side fashion. Similarly, with a sufficiently large display, a user would be able to display and view a number of execution windows simultaneously on one display.

In addition to having increased size, it would also be an improvement to have a display system that can be oriented about a computer user so as to provide the user with peripheral views of three-dimensional display graphics -- such as would be found in a virtual reality-type application -- and thereby provide the viewer with a more realistic and

"immersive" viewing environment. At the same time, the display system should be capable of simultaneously displaying both 3D and 2D applications, thereby increasing the flexibility of the display and the productivity of the user. Moreover, it would be desirable if the video display system could be implemented on computer systems using currently available graphics processor electronic technology, and currently available operating system technology.

SUMMARY AND OBJECTS OF THE INVENTION

The foregoing problems in the prior state of the art have been successfully overcome by the present invention, which is directed to a multiscreen display system and method. The multiscreen display system of the present invention is comprised of a plurality of interconnected video displays. The video displays are each positioned with respect to one another in a spatially adjacent manner so that each of the individual display screens together function as a single contiguous display screen. Moreover, in a preferred embodiment, the display screens are interconnected and arranged in a manner so as to form a concave shape about the computer user.

The display system is connected to and can be used in conjunction with a standard personal or desktop computer, and can be used with currently available multi-tasking operating systems. Moreover, the system can be used to display both two dimensional applications and complex three dimensional applications. Individual displays, or alternatively, groups of individual displays within the display system, are electrically connected to the desktop computer via graphics processor electronics positioned within the computer housing.

Stored within an appropriate computer data storage area is "screen position data," which contains the position/location of each of the individual display screens within the display system. Preferably, the screen position data signifies each of the individual screen positions in relation to an assumed viewer position. This data can then be used by a graphics application executed on the computer to provide a more realistic "virtual" image display, and to determine adjacency for 2D applications.

A computer executable program method is also disclosed for use in connection with the display system. The program method is directed to the program steps that can be utilized by the graphics processor electronics to send image data to the appropriate display screen within the display system. Moreover, the executable program method allows for use of specialized display system data packets as well as data packets that take advantage of existing graphics processor subroutine functions libraries.

Program methods for implementing complex three dimensional visualization software applications in connection with the multiscreen display system are also disclosed. That program method utilizes the screen position data set in a manner so as to provide the ability to generate extremely realistic, immersive virtual graphics images on the display system.

In operation, the multiscreen display system provides the ability to have a large display screen size for displaying numerous application windows and large amount of data and graphics. Moreover, the display system provides a user with the ability to interact with both complex three dimensional graphics applications and two dimensional office-type productivity applications at the same time and on the same screens. Finally, the concave arrangement of the system provides a user with peripheral vision of image graphics, and results in an immersive field of view by the computer viewer.

BRIEF DESCRIPTION OF THE DRAWINGS

In order that the manner in which the above-recited and other advantages and objects of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawing depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

Figure 1 is a perspective view of one example embodiment of the multiscreen display system of the present invention;

Figure 2 is a perspective view of another embodiment of the multiscreen display system of the present invention;

Figure 3 is a perspective view of yet another embodiment of the multiscreen display system of the present invention;

Figure 4 is yet another perspective view of the multiscreen display computer display system of the present invention;

Figures 5A-5G are perspective views of a fold-down version of a multiscreen computer display;

Figures 6A-6D are each perspective views showing various embodiments of interconnection schemes between the individual display panels of the multiscreen display system of the present invention;

Figures 7A-7B are functional block diagrams illustrating certain of the functional components used in the multiscreen display system;

Figure 8 is a functional block diagram illustrating the functional components used in one example of an electronic graphics processor used in the multiscreen display system of the present invention;

Figures 9A-9C illustrate the manner by which screen position data is calculated as well as an alternative screen arrangement that can be used in the multiscreen display system;

Figure10 is a flow chart illustrating one presently preferred series of program steps for use in connection with the multiscreen display system of the present invention;

Figure 11 is a second program flow chart illustrating presently preferred program methods for use in connection with the multiscreen display system.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention contemplates both systems and methods for implementing a multiscreen computer display environment. The description that follows describes various embodiments of the invention, and is broken down into three sections. The first section describes examples of the overall system by making reference to figures that illustrate the various embodiments of the a multiscreen display system and the general computing environment in which the system may be implemented. Additional details pertaining to the display system are described by referring to functional diagrams that illustrate examples of some of the functional blocks that can be used to implement one preferred embodiment of the invention.

The second section of the discussion is directed to a description of examples of the "screen position data" that is used in connection the implementation of certain of the embodiments of the invention. In addition, examples of preferred methods by which the screen position data is calculated are provided by making reference to certain figures.

The third section of the detailed description is directed to a description of program methods for use in connection with the operation and use of the multiscreen display system. Preferably, methods will be implemented by way of computer executable instructions stored on a computer program product. The instructions will be primarily described by making reference to various program flow diagrams.

It will be appreciated that the following description, as well as the accompanying figures and diagrams, should not be construed as limiting of the present invention's scope. Instead, the description and drawings should be viewed as illustrating an example of a presently understood preferred embodiment of the invention.

I. THE SYSTEM

Reference is now made to Figure 1, which illustrates one presently preferred embodiment of a multiscreen display system constructed in accordance with the teachings of the present invention. In the embodiment illustrated, a multiscreen display, designated generally at 10, is connected to a general purpose personal, or desktop computer, designated generally at 12. Those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including without limitation multi-processor systems, network PCs, minicomputers, and the like. As is shown, the desktop computer 12 includes a standard computer keyboard 14 that is operatively connected to the computer system chassis 13. Other user input devices could also be connected, such as a mouse, or similar pointing device, such as a joystick, trackball, touchscreen, etc. The chassis 13 is what houses the processing unit, computer system bus, power supply, and related computer electronics of the computer. Also positioned within the chassis is graphics processor electronics, designated at 16, which is used in connection with the control and operation of multiscreen display 10. In addition, a personal computer chassis 13 will typically include at least one type of computer storage device, such as, for instance, at least one hard disk drive, a magnetic floppy disk drive, an optical drive, etc. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions and data for the personal computer 12.

Personal computer 12 further includes a CPU/processing unit (not shown), a computer system memory area (such as read only memory ROM and random access memory RAM), and a computer system bus (I/O bus) (designated at 40 in Figure 8, described below), which interconnects each of the system components with the CPU/processing unit. The system bus may be comprised of any type of bus structure that is commonly used in various computer architectures and implementations.

In the embodiment illustrated, the multiscreen display 10 is comprised of a plurality of mutually connected thin display panels or screens, which are designated at 18a-n. These individual display panels 18a-n can be comprised of any electrically active technology including for instance, liquid crystal (including active and passive matrix), electro-luminescent, plasma, directed emission cathode ray, or any similar type video display technology. Also, different display technologies can be mixed within the system 10.

In the example embodiment shown in Figure 1, the screens 18a-n are mutually connected in a generally spatially adjacent manner, and are arranged and oriented in a roughly concave shape around the computer user. Preferably, the mutually connected

screens 18a-n are supported by an interconnecting framework of horizontal 20 and vertical 22 support rods or beams. The framework is oriented so as to rest on a standard height desktop 24. Preferably a support brace 26 attaches the framework of display screens 18a-n to the desktop 24 and provides additional support thereto.

While the embodiment illustrated in Figure 1 shows one preferred arrangement of the display system 10, it will be appreciated that other physical configurations could also be used. For instance, in the display system 10' of Figure 2 the individual display screens 18a-n' are curved, and not planar. Also, the host computer 12', cabling, graphics circuits 16' and other components are integrated within a supporting framework housing 15. Also, a hinged portion of the supporting framework 17 opens to allow access to the internal electronics of the integrated host computer 12'. Figure 3 illustrates an embodiment wherein the display system 10" functions as a docking station for a portable computer 12". Once docked, the laptop screen 19 works in conjunction with the other individual display screens 18a''-18n'' within the display system 10". In Figure 4, the central portion of the screen array is omitted to allow a standard CRT-type video monitor 21 to be used. The position of the CRT monitor 21 relative to the viewer can be added to the screen position data (discussed below), thereby allowing it to function within the display system 10'''. In the embodiments of Figure 5A-G, a fold-up multiscreen display system is illustrated in various stages of retraction. In that embodiment, hinges 23 are used to attach adjacent screens 18a-n'''', thereby allowing the screens to be folded compactly. In Figures 9A-9C, an embodiment of a multiscreen display system using polygonal shaped screens is shown. Moreover, this embodiment illustrates how individual screens can be arranged so as to be spatially adjacent, but in a non-contiguous manner.

Referring again to the embodiment of Figure 1, as will also be discussed in further detail below, each of the individual display screens 18a-n are electrically interconnected to the graphics circuitry 16 that is installed within the desktop computer 12. Preferably, this interconnection is accomplished by way of electrical display cables, which are designated at 28. Stored within an appropriate computer data storage area (*e.g.*, magnetic storage, RAM, ROM) is "screen position data," which contains the position of each of the individual display screens 18a-n. The content, the format, and the preferred procedure for calculating the screen position data will be described in further detail at section II below. In general however, the screen position data preferably denotes each of the individual screen positions in relation to an assumed viewer position, such as would correspond with the eye level of the computer user 30. This screen position data can then be utilized by an application executing on the computer 12, such as a complex

visualization, virtual reality application, or similar 3D graphics application. The position data can be used, for instance to construct contiguous, wrap-around computer graphics images in three dimensions across part, or all of the display screens. Similarly, two-dimensional applications, such as would commonly be executing within windows-based operating system application windows, could also utilize the screen position data to allow, for instance, the screen cursor to move between contiguous windows on the display screen 10. Also, two dimensional text windows could span across adjacent screens, thereby providing much larger display areas for a windowed application.

The visible color of individual pixels of each of the distinct screens 18a-n are controlled by a row and column arrangement of electrical conductors. It will be appreciated that various methods could be used to facilitate the electrical interconnection of columns and/or rows of adjacent screens 18a-n to create electrical groupings of screens. For instance, the number of display cables, and graphics control circuits (designated at 28 and 16 in figure 1), required is determined by the extent of electrical grouping of these individual screens 18a-n Figures 6A-6D illustrates examples of the various embodiments of the electrical interconnection schemes that can be used in connection with embodiments of the current invention. For example, in Figure 6A each of the individual display screens 18a-n will have a separate cable (28 in Figure 1) and a separate graphics control circuit (16 in Figure 1) associated therewith. In contrast, the embodiments illustrated in figures 6B and 6C utilize separate electrical cable conductors, designated at 32, to interconnect adjacent screens to form groups of screens. These approaches would each require the use of fewer individual cables and graphics control circuits (i.e., a separate cable and graphics control circuit for each group of screens). The interconnection scheme shown in figure 6D utilizes an alternate manufacturing process, wherein the electrical conductors 32 are formed as mutually adjacent electrical plugs and receptacles so that adjacent screens can be plugged into one another in an integrated manner, or manufactured as a single unit without connectors.

The display cables (28 in figure 1) that are used to interconnect the graphics control circuitry 16 to the individual display screens 18a-n will carry either analog or digital signals, depending on the particular type of display electronics being used. These alternatives are illustrated in functional block form in Figures 7A-7B. As is well known, digital control signals require less electronics, but are limited in length-of-travel to about one meter. Thus, use of digital display screens would be preferred where the display screens are spatially located so as to be closer to the computer (*See*, for instance, the alternative embodiments illustrated in Figures 2, 3 and 5). Alternatively, analog control signals require additional circuitry, but the length-of-travel is extended to about two

meters. As such, this approach would be preferred where the display screen is located a further distance from the host computer, as for instance, with the embodiment illustrated in Figures 1 and 4.

Reference is next made to Figure 8, which is a functional block diagram illustrating the basic functional components included within the graphics control circuitry (16 in Figure 1). The graphics control circuitry, which is designated generally at 16 in Figure 8, is preferably implemented with known circuit components on a printed circuit board, which is in turn received within the computer chassis 13. The printed circuit board includes appropriate connectors for attachment to the respective display cable(s), and for attachment to the host computer system I/O bus, which is designated at 40 in Figure 8. It will be appreciated that various other physical embodiments of the graphics control circuit board hardware 16 could be used. For instance, in the embodiment of Figure 1, the circuit is implemented as a standard sized adapter printed circuit card, wherein each circuit card interfaces with one or more display screens. Each card is inserted into a system I/O bus interface slot provided on the host computer motherboard in a manner well known in the art. Such graphics cards, which each drive one or more independent screens, are currently available from a number of different manufactures. With this embodiment, circuitry for interfacing with the computer system I/O bus, and any associated custom control circuitry (discussed below) is required on each separate physical board. In an alternative embodiment, a single circuit board containing the graphics control circuitry for an entire multiscreen display system can be used. With this approach, a bus extender board could be used to connect the single card to the physical connector for the system I/O bus of an unmodified host computer motherboard. Here, only one bus interface circuit and custom control circuit is required. This approach is advantageous because it uses fewer interface bus expansion slots on a computer motherboard. Alternatively, a custom host computer motherboard having the graphics control circuitry implemented thereon for an entire display system could be used. This approach would provide performance advantages because the communications between the host processor/CPU and the graphics circuitry would occur over the high-speed memory bus as opposed to the lower speed computer system I/O bus. Also, the bus interface circuit is eliminated completely, and a custom control circuit could be replaced by a memory controller circuit which also services other components on the motherboard.

Preferably, the graphics control circuitry 16 is comprised of graphics processor circuit means for causing image data to be displayed on a corresponding display, and multi-screen controller means for forwarding image data from an application executing on the computer to the appropriate graphics processor circuit means. By way of example

and not limitation, the graphics processor circuit means is comprised of a 20 graphics processor 44 and/or a 30 graphics processor 46 circuit, discussed below. Also, the multi-screen controller means is comprised, for example, of a custom control circuit 52, which is also discussed in further detail below.

With continued reference to Figure 8, it is shown that in a preferred embodiment certain of the functional components of the graphics control circuitry 16 are interconnected with a high-speed graphics processor bus 42. In the illustrated embodiment, a 2D graphics processor 44a-n, a 3D graphics processor 46a-n, a video processor/decoder 48a-n, a memory block 50, and a custom control circuit 52 are all interconnected by way of the high speed graphics bus 42. As indicated, a single custom control circuit 52 is required for each physical circuit board that interfaces with the computer I/O bus 40. Also, in the example embodiment, display control circuitry, designated in the dotted box 54a-n, is required for each independent screen 18a-18n, or screen group (Figure 6). In addition, at least one memory block 50 is required.

As previously noted, each of the individual displays 18a-18n receive electrical timing and control signals via a display cable 28a-28n via a corresponding display controller circuit 56a-n. The display controller circuit 56a-n operates a manner that is well known in the art by outputting the proper electrical timing and control sequences required by the corresponding display screen 18a-n. As is shown, an intervening digital-to-analog converter circuit 58a-n is used where the cable 28a-n carries analog signals. Of course, as previously described and discussed in connection with Figure 7, the D-to-A circuit 58 is not needed if digital signals are utilized. Display controller integrated circuits that can be used to implement the function denoted at 56a-n are well known in the art and are available from numerous manufacturers. Such integrated circuits commonly provide both analog and digital outputs.

In operation, the display circuit 56a-n periodically reads the currently displayed image from a display memory denoted at 60a-n. The contents of the display memory 60a-n include a representation of the image that is currently being displayed on the display screen 18a-n. In a manner that is well known, the contents of the display memory 60a-n, and the corresponding image on the display screen 18a-n, are periodically modified by the 2D graphics processor 44a-n, the 3D graphics processor 46a-n, and the video decoder 48a-n circuits. Display memory integrated circuits that can be used to implement the function represented at 60a-n are well-known in the art and are available from more than one manufacturer. Also, such integrated circuits are slightly faster than other types of computer memory, and are typically sized so as to accommodate the resolution and color depth of the corresponding display screen being used. Moreover,

dual-ported display memory integrated circuits, which allow for the stored image to be both read and modified at the same time, are also available from a variety of different manufacturers. Also, 2D processors or 2D accelerator integrated circuits, as well as 3D accelerator or 3D processor integrated circuits, which can be used to provide the functions illustrated at 44a-n and 46a-n respectively, are available from a number of different manufacturers. In addition, video processor or video decoder integrated circuits, which function to decompress video images in real-time, are also available from different manufacturers and can be used to provide the function illustrated at functional box 48a-n.

The 2D processor circuit 44a-n, the 3D processor circuit 46a-n, and the video processor circuit 48a-n each accept commands and data from the custom control circuit 52, and reference information contained within the memory block 50, via the high-speed graphics processor bus 42. This command, data and reference information is collectively used to modify the image represented in the display memory 60a-n.

In the embodiment illustrated in Figure 8, a significant level of parallel graphics activity can be achieved due to the use of independent graphics processing circuits 54a-n for each display screen 18a-n, or display screen group. As will be discussed in further detail below, to attain such parallel activity, software commands and data are given to, for instance, the graphics circuitry associated with functional block 54a, and while that circuitry is processing the data, the graphics circuitry associated with functional block 54b-n can be provided with software commands and data. Again, while those graphics circuits are processing, other graphics circuits can also be given respective commands and data.

With continued reference to Figure 8, a setup processor circuit, designated at 70, functions so as to assist the graphics processors 44a-n, 46a-n by optimizing certain of the relatively static image parameters, including lighting and texture data values, so as to provide for higher speed reference and performance by the graphics processors. Not all commercially available 3D graphics processor integrated circuits have a companion setup processor integrated circuit 70. Thus, the use of this particular circuit is dependent upon the particular 3D graphics processor that is being utilized.

The custom control function designated at 52 controls the flow of data between the various functional circuits that make up the graphics processor electronics 16. The general function provided by the circuit 52 is to accept data from the host computer processor (not shown) via the host computer system I/O bus 40. The circuit 52 will then either act upon the data, or pass it along to a destination display control circuit 54a-n, which can be selected via assertion of an appropriate signal by the multiplex logic 73. It will be appreciated that various programmable circuits could be used to provide this

particular bus interface function, which are available from a number of manufacturers. Examples of presently preferred programming methods to implement certain of the functions of the circuit will be discussed below. The function provided by the custom and control circuit 52 can be with a custom circuit in the form of an application-specific integrated circuit (ASIC). Alternatively, the function could be implemented with a programmable microcontroller. The read-only program memory module 72 may be integrated within the device itself, or may be comprised of an external memory module device. The program memory 72 is used to store the programmable instructions used to control the operation of the device and to implement the custom control function. In addition, bus interface logic 74 may also be implemented within the custom control circuit 52. The bus interface logic circuitry 74 provides the proper timing and control signals to interface with the computer system I/O bus 40, and passes data between the bus 40 and the rest of the custom control circuit 52. Again, this particular function could be implemented within the ASIC/microcontroller, or could be implemented by way of a separate, standalone circuit module, of which there are a variety available from a number of manufacturers. The specific programming logic required to interface with a computer interface system I/O bus will depend on the bus used, and is well known by those of skill in the art.

The memory block 50 is preferably implemented as a single, random access memory area which is implemented and resides on the same physical board as the rest of the graphics control circuitry 16. This approach allows for high-speed access by the graphics processors 44, 46 and the video decoder 48, which also reside on the board. Also, increasing the size of this memory location correspondingly increases the number of texture maps (stored in texture memory 80) and the amount of 3D rendering information available in polygon memory 81, setup memory 83 and parameter memory 85, which also increases display performance. Larger texture memory area 80 also allows for more detailed texture maps which increases the richness and detail of the displayed images.

II. THE "SCREEN POSITION DATA" STRUCTURE

As already noted, in order to best utilize the multiscreen display system 10 described above, a data structure referred to as the "screen position data" is created and stored within an appropriate computer storage location, such as computer disk storage. This data is then made available to application programs executing on the desktop computer 12 so that the display system 10 can be used in a suitable manner.

Figure 9 is illustrative of a preferred process for creating the screen position data set. An assumed viewer location, designated at 30, which is preferably located at a point central to the concave, or similarly oriented, display system 10 is identified. For instance, the assumed viewer location can be calculated based on an average stature for a typical computer user, that is seated on an average height chair with the display system 10 resting on a standard height desktop. For nonstandard desktop applications, similar methods can be used to determine an assumed viewer position.

Relative measurements for each display screen 18a-n location within the display system 10 are then made from this assumed viewer position. For instance, the assumed viewer position is identified as the 0,0,0 position in an X, Y, Z coordinate grid, and measurements similar to that which would be obtained by a three-dimensional coordinate measurement machine can be taken to identify the various screen 18 locations. Preferably, the locations of each corner of the active area of each display screen 18a-n with respect to the assumed viewer position are calculated. These distance values are then sorted and stored in a clockwise order starting from the top-most (in the positive Y coordinate direction) edge of each display screen 18a-n. These values are stored digitally within computer storage as screen position data, examples of which are shown below in Table A. Preferably, the screen position data set also includes the units of measure being used, a screen identifier and screen specifications/characteristics which are significant to the rendering of the images. For instance, the data can include the resolution of the screen, the color capabilities, etc. The specific format and content of the screen position data can then be communicated to a software developer, who can then use the data to determine view volumes and other image rendering aspects/characteristics to be displayed on the display system 10.

It will be appreciated that a number of reference coordinate systems exist for such measurements, including polar, and non-right-handed. Any such measurement system could be used as long as the method and units are clearly communicated and usable by the program developer.

TABLE A

```
# Screen Position Data File
#------------------------------

units=inches

screen = "optional name of screen 1"
hres/vres/color/gama = 800 600 16 0
x = +004.00 y = +006.00 z = -018.00
x = +008.00 y = +000.00 z = -018.00
x = +004.00 y = -006.00  z = -018.00
x =  -004.00 y = -006.00  z = -018.00
x =  -008.00 y = +000.00 z = -018.00
x =  -004.00 y = +006.00 z = -018.00

screen = "optional name of screen 2"
hres/vres/color/gama = 800 600 16 0
x = +004.00 y = -006.00 z = -018.00
x = +004.00 y = -011.64 z = -015.95
x =  -004.00 y = -011.64 z = -015.95
x =  -004.00 y = -006.00 z = -018.00

screen = "optional name of screen 3"
hres/vres/color/gama = 800 600 16 0
x = -004.00 y = +006.00 z = -018.00
x = -008.00 y = +000.00 z = -018.00
x = -013.68 y = +000.00 z = -012.32
x = -012.49 y = +006.00 z = -009.51

screen = "optional name of screen 4"
hres/vres/color/gama = 800 600 16 0
x = +012.49 y = +006.00 z = -009.51
x = +013.68 y = +000.00 z = -012.32
x = +008.00 y = +000.00 z = -018.00
x = +004.00 y = +006.00 z = -018.00

screen = "optional name of screen 5"
hres/vres/color/gama = 800 600 16 0
x = -012.00 y = +003.00 z =  -004.00
x = -012.00 y =  -003.00 z =  -004.00
x = -012.00 y =  -003.00 z = +004.00
x = -012.00 y = +003.00 z = +004.00

screen = "optional name of screen 6"
hres/vres/color/gama = 800 600 16 0
x = -008.00 y = +009.93 z = +000.00
x = -012.00 y = +003.00 z =  -004.00
x = -012.00 y = +003.00 z = +004.00

screen = "optional name of screen 7"
hres/vres/color/gama = 800 600 16 0
x = +008.00 y = +009.93 z = +000.00
x = +012.00 y = +003.00 z = +004.00
x = +012.00 y = +003.00 z =  -004.00

screen = "optional name of screen 8"
hres/vres/color/gama = 800 600 16 0
x = +012.00 y = +003.00 z =  -004.00
x = +012.00 y =  -003.00 z =  -004.00
x = +012.00 y =  -003.00 z = +004.00
x = +012.00 y = +003.00 z = +004.00
```

In a preferred embodiment, the screen position data may be supplemented with an "adjustment data set." The adjustment data set contains correction values that are used to adjust the assumed viewer position data values to correspond with an actual viewer position. This may be due to differences in the height of computer user viewer, the chair height, and desk height. This data set may be manipulated each time a user invokes an application. For instance, a user may manually enter in relevant adjustment data, by taking measurements with respect to an arbitrary reference point on the display system 10. Alternatively, the actual head position of a user and/or the display system could be

determined by electronic sensors that are integrated within the framework of the display system 10, and the adjustment data set could then be calculated automatically. However, it is anticipated that in most cases, no adjustment is required since minor variations from the assumed viewer position are largely unnoticeable.

III. THE METHOD

Another aspect of the present invention pertains to methods for controlling and utilizing the multiscreen display system 10 described above. Embodiments within the scope of the present invention may include computer readable media having executable instructions. Such computer readable media can be any available media which can be accessed by a general purpose personal computer. By way of example, and not limitation, such computer readable media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired executable instructions and which can be accessed by a general purpose computer. Combinations of the above should also be included within the scope of computer readable media. Executable instructions comprise, for example, instructions and data which cause a general purpose computer to perform a certain function or group of functions.

As is well known, the manufacturers of 2D, 3D and video decoder integrated circuits (elements 44, 46 and 48 respectively in Figure 8) ordinarily provide an ability to control the devices with software in the form of a library of graphics subroutines that can be accessed with subroutine calls made via a predefined, application programming interface (API). In the preferred embodiment, the ability to take advantage of the various functions provided via these graphic subroutine libraries is provided. Thus, function calls that are specific to the multiscreen display 10 can be mixed with function calls to graphic subroutines specific to a particular integrated circuit (*i.e.*, 2D, 3D and/or video decoder) included within the system.

Following is an example of the sort of source code that can be implemented as executable instructions that run on the host computer 12, and which addresses multiple display screens 18 without requiring alterations to graphics processor specific application programming interface (API) libraries.

```
# 3D Visualization Software -- Source Code Example
# ------------------------------------------------------------

include:    Graphics Processor Specific API Functions
include:    Multi-Screen Specific API Functions contains
            the "setScreen()" function

    for i = 1 to Number-of-Screens

        setScreen(i);   # This is a multiscreen API function call
        .........

        Graphics Processor Specific function calls;
        .........

    loop;
```

This inter-mixing of library calls creates a sequential data stream to appear on the host computer system I/O bus (40 in Figure 8). For instance, in the above example, a screen selection data packet reaches the custom control circuit 52 from the system I/O bus 40. The interface logic receives the screen selection packet data packet, directing the multiplex logic 73 portion of the circuit to assert a chip enable, or chip select line for the graphics processor associated with screen "i." All other graphics processors are simultaneously de-activated from receiving input by the same logic. Any subsequent graphics processor specific data packets, or function calls defined by the selected graphics processor's library, are then passed unmodified by the custom control circuit 52 onto the graphics processor bus 42, and are then accepted by the single enabled graphics processor associated with screen "i." This continues until another screen selection data packet is received by the multiplex logic 73 portion of the custom control circuit 52, which then selects/enables a new graphics processor 54a-n. Alternatively, the graphics processor specific API library function calls could be modified to require, for instance, a screen number argument within a parameter list, and a screen number within each data packet sent on the host computer system I/O bus 40. This identifier could be used by the custom control circuit to direct the data packets to the appropriate graphics processor circuit 54a-n.

Reference is next made to Figure 10, which is a program flow chart illustrating one presently preferred embodiment of the program steps that can be used to control the operation of the programmable multiscreen custom control circuit 52. The series of executable instructions begins at program step 100, where the circuit 52 obtains a data packet from the host computer system I/O bus 40. Once a packet is captured, at program step 102 the circuit 52 checks to see if it is a multiscreen specific data packet. If not, the

process continues at program step 104, and the packet is passed unaltered to the high speed graphics processor bus 42, from which the then enabled graphics processor circuit 54a-n will receive and process the packet accordingly. The custom control circuit 52 will then return to program step 100 and wait for the next data packet.

If however at program step 102 it is determined that the packet is in fact a multi-screen specific data packet, then the program steps will proceed as indicated at schematic line 106, at which point a series of queries will be performed so as to ascertain which function is to be performed. Thus, processing will begin at program step 108 where the circuit 52 will determine if the data packet corresponds to a set screen command. If so, at program step 110 the custom control circuit 52 will enable the particular graphics processor 54 that has been selected. This is accomplished, for instance, by asserting a corresponding enable signal via the high speed graphics bus 42. If the data packet does not include a set screen command, then processing will proceed to program step 112, where it is determined whether the data packet includes an inquiry command requesting that the currently enabled display screen be identified. If so, at program step 114 the custom control circuit 52 will return a packet to the host CPU system I/O bus 40 that contains the currently enabled display screen. If the data packet does not include such an inquiry command, processing will continue at program step 116. That program step corresponds to any other valid multiscreen data packet commands that may be present. If a valid multiscreen command is present within the data packet, processing continues at program step 118, and the corresponding command is processed by the custom control circuit 52. If however, no valid multiscreen command is contained within the data packet, program step 120 is performed, and an appropriate error message is returned to the host CPU system I/O bus 40. Once the data packet has been queried for a valid multiscreen command, and the command has been identified and processed, or the appropriate error message has been returned, processing will continue at program step 100 at which point the custom control circuit 52 will again wait for another data packet from the host CPU system I/O bus 40.

Reference is next made to Figure 11, which illustrates the presently preferred program method steps corresponding to the manner by which a three-dimensional visualization software application would utilize the multiscreen display system 10. Generally, these program steps would be performed by the host computer 12 processor CPU. Beginning at program step 200, an application would begin by reading the screen position data set designated at 202, an example of which was illustrated and discussed in connection with Table A above. Once that data has been obtained, the processor

would proceed to the series of executable steps corresponding with block 204, where the viewer position adjustment data set, designated at 206 and described above, is read.

Once the screen position data set and viewer position adjustment data set have been read, the processor proceeds to step 208, where it calculates, based on the adjusted screen position data, the three-dimensional view volumes for each screen 18a-n. The proper portion of the surrounding "virtual" image that the viewer would see, *i.e.*, as if they were looking through a window at the same location, is then displayed. Generally, if four lines are drawn in space from the viewer position to each corner of a single rectangular screen, a pyramid shape is created. When this pyramid is mathematically extended into a virtual world, it defines the portion of the virtual world that must be displayed on the screen. The mathematical description of this extended pyramid is called a view volume. Thus, a front screen would contain a view of the virtual world to the north, a left screen would contain a view of the virtual world to the west, and a right side screen would contain a view of the virtual world to the east. In this way, the views change as the viewer moves and turns within the virtual world. An encompassing view volume, which is the view volume of the entire display system 10 rather than a single screen 18a-n, can also be calculated to speed certain software operations.

At program step 210, various initialization activities can be performed, which include resetting graphics circuitry, clearing the keyboard input buffer, clearing all display screens, etc.

The host computer 12 processor then enters an event loop, referred to as a primary event loop for 3D visualization software and designated at 212, and waits for some action or event to occur. Based on the particular event detected, the program will perform a corresponding sequence of instructions. If events occur in rapid succession, each event will be queued within memory and handled in the order that they occurred.

For instance, if the event is a movement of the viewer within a three-dimensional virtual world, then there is a very high likelihood that all screens must be updated. Thus, at program step 214, when a move viewpoint event is detected, a screen update loop, beginning at program step 216 is invoked. The series of program steps in that particular loop proceed to update the image that is currently shown on all display screens 18. The graphics application software can require that this particular loop finish so that all screens get updated before accepting the next event, or the software can be executed so as to exit the loop prematurely and start again if another move viewpoint event occurs. Alternatively, the central display screens within the display system may complete the update loop and be updated, while the updates to the peripheral display screen are interrupted. Computational speeds, image complexity, and programmer preference are

some of the variables involved in determining the most appropriate way to write three-dimensional visualization software that utilizes multiple display screens in this manner.

Program step 222 corresponds to the circumstance where a localized event occurs, such as the movement of a single object within the image. In this circumstance, it may be advantageous to do the mathematical calculations to determine which screens need to be updated. By comparing the location (and size and distance) of the localized event with the encompassing view volume, it can be determined if the localized event appears within view of any of the screen view volumes. If not, no screen updates are required, as shown at program step 224. If however the localized event is contained within the encompassing view volume, then all screens can be updated, or alternatively, only those screens in which the event appears can be updated, as is shown at program steps 226-232. Again, such approaches are highly application dependent and left up to the author of the 3D visualization software.

Program step 234 indicates that any other relevant 3D visualization events can also be detected and responded to, again depending on the particularities and requirements of the specific 3D visualization software that is being used.

In operation, when a 3D graphics application is displayed, a virtual world will appear on all or part of the display screens 18a-n so as to provide the viewer with the feeling of immersion imparted by the extended field-of-regard offered by the display system 10. The viewer is free to look around this virtual world through head and eye movements without affecting the stability of the image. Motion of the viewer within this virtual world is accomplished with input devices including keyboard, mouse, joystick, trackball, touch screen, treadmill, stationary bicycle, rowing machine, or any other input device which provides a roughly stationary head position.

If virtual world viewing is not required, the display system 10 also provides sufficient resolution to display 2D textual application windows. For instance, a standard windowing operating system allows the viewer to launch applications which create 2D textual application windows on the display system. The viewer is free to move these windows to any screen and may expand them to fill that screen. Application windows may also be expanded to span across multiple screens within the display system, thereby allowing the entire contents of an application window to be displayed at once, or multiple pages of a document to be displayed side-by-side. Similarly, the mouse-cursor can move across all screens in a contiguous fashion. In this expanded-desktop mode of operation, the viewer is able to place an email reader on one screen, a real-time stock ticker on another screen, an Internet browser on another screen, a to-do list on another screen, an appointment calendar on another screen, a word-processing document on another screen,

and a database access window on another screen. In this way, all windows are simultaneously visible for reference, without need to rearrange or toggle between application windows. The viewer may also configure this multiscreen system for video conferencing, wherein each conferencee's image is given a separate non-overlapping screen. A camera can be integrated within the supporting framework to provide the viewer's image to others.

Another viewing alternative provided by the display system 10 is the viewing of both 3D virtual worlds and 2D textual application windows simultaneously. In this mode of operation, the desktop background effectively becomes the dynamic image of the virtual world, and the 2D application windows appear in front of the 3D scene. In this mode the viewer may position a small email window, for example, near the periphery of viewer's vision. The user can then easily monitor the status of an incoming email, for instance, while simultaneously executing a virtual reality-type application on the other screens within the display system 10. This mode is also useful for providing information and controls relating to the virtual world currently being viewed. For instance, in game applications, such 2D windows can be used to display score, status, and options. In an exercise machine application, heart rate, elapsed time, and level of difficulty can be displayed in 2D windows. In a remotely operated vehicle application, fuel level, altitude, bearing, and speed can be displayed in 2D windows.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed is:

1. A multi-screen display system for use by a computer, the display system comprising:

a plurality of display screens arranged in a spatially adjacent manner so as to together form a single contiguous display screen;

screen position storage means for containing screen position data that is representative of the relative position of each of the plurality of display screens and that is measured with respect to an assumed position of a user of the computer;

a graphics processor circuit that is operatively associated with at least one display screen and that is capable of causing image data to be displayed on the associated display screen; and

multi-screen controller means for forwarding image data from an application executing on the computer to the graphics processor circuit associated with the appropriate display screen based on the screen position data.

2. A multi-screen display system as defined in claim 1, wherein the plurality of display screens are arranged in a manner so as to form a contiguous display screen that is concavely shaped and oriented about a user of the computer.

3. A multi-screen display system as defined in claim 1, wherein the image data received by the graphics processor circuit can be used to generate an image on the associated at least one display screen in a manner so that a single continuous image is produced on the contiguous display screen formed by the plurality of display screens.

4. A multi-screen display system as defined in claim 1, wherein the graphics processor circuit includes at least one of the following: a two-dimensional (2D) graphics processor circuit and a three-dimensional (3D) graphics processor circuit.

5. A multi-screen display system as defined in claim 4, wherein the graphics processor circuit further comprises:

a display memory for storing image data that is to be displayed on the associated display screen; and

a display driver controller circuit that is capable of outputting predefined electrical timing and control sequence data for causing the image data contained in the display memory to be displayed on the associated display screen.

6. A multi-screen display system as defined in claim 1, wherein the multi-screen controller means comprises a multiplex logic circuit that is capable of receiving image data from an application executing on the computer and forwarding the data to a selected graphics processor circuit based upon the contents of the image data.

7. A multi-screen display system as defined in claim 6, wherein the multi-screen controller means further comprises computer storage memory for containing at least one of the following parameters: texture maps corresponding to and further defining the image defined by the image data; and rendering information corresponding to and further defining the image defined by the image data.

8. A multi-screen display system as defined in claim 1, wherein the plurality of display screens are comprised of thin-panel computer display screens.

9. A multi-screen display system as defined in claim 1, wherein the screen position storage means further contains a unique screen identifier for each display screen within the display system.

10. A multi-screen display system as defined in claim 1, wherein the screen position storage means further contains an adjustment data set, the adjustment data set containing correction values that can be used to adjust the screen position data values to reflect an actual position of a user of the computer.

11. A multi-screen display system for use by a computer, the display system comprising:

a plurality of thin-panel computer display screens arranged and physically interconnected in a spatially adjacent manner so as to together form a concavely shaped single contiguous display screen for use by the computer;

a memory storage area operatively connected to the computer that contains screen position data that is representative of the absolute position and location of each of the plurality of thin-panel computer display screens relative to an assumed position of a user of the computer;

a three-dimensional graphics processor circuit that is operatively associated with each separate thin-panel computer display screen and that is capable of generating three-dimensional images on the associated thin-panel display screen;

a two-dimensional graphics processor circuit that is operatively associated with each separate thin-panel computer display screen and that is capable of generating two-dimensional images on the associated thin-panel display screen; and

multi-screen controller means for forwarding three-dimensional image data from an application executing on the computer to the three-dimensional graphics processor circuit associated with the appropriate thin-panel display screen based on the screen position data, and for forwarding two-dimensional image data from an application executing on the computer to the two-dimensional

graphics processor circuit associated with the appropriate thin-panel display screen based on the screen position data.

12. A method for distributing image data to a plurality of computer display screens that are connected to a computer so as to form a single contiguous display screen, the method comprising the following steps:

(a) storing in a memory location screen position data that is representative of the location and position of each of the plurality of computer display screens measured with respect to an assumed position of a user of the computer;

(b) receiving an image data packet containing image data for display on one of the plurality of display screens;

(c) based upon the contents of the image data packet and upon the screen position data, forwarding the image data packet to a graphics processor associated with the correct display screen;

(d) displaying by the graphics processor an image on the corresponding display screen in accordance with the contents of the image data packet; and

(e) repeating steps (b)-(c) so that a single continuous image is produced on the single continuous display screen formed by the plurality of display screens.

13. A method as defined in claim 12, further comprising the steps of:

calculating an adjustment data set, the adjustment data set containing screen position correction values; and

adjusting the screen position data values based on the adjustment data set so that the screen position data values reflect an actual position of a user of the computer.

14. A method as defined in claim 12, wherein at least one two-dimensional image is displayed on at least one display screen at the same time as at least one three-dimensional image is displayed on at least one of the other display screens within the display system.

15. A method as defined in claim 12, wherein images on at least one display screen are updated in accordance with corresponding actions of the computer user.

16. A computer-readable medium having computer-executable instructions for performing the steps recited in claim 12.

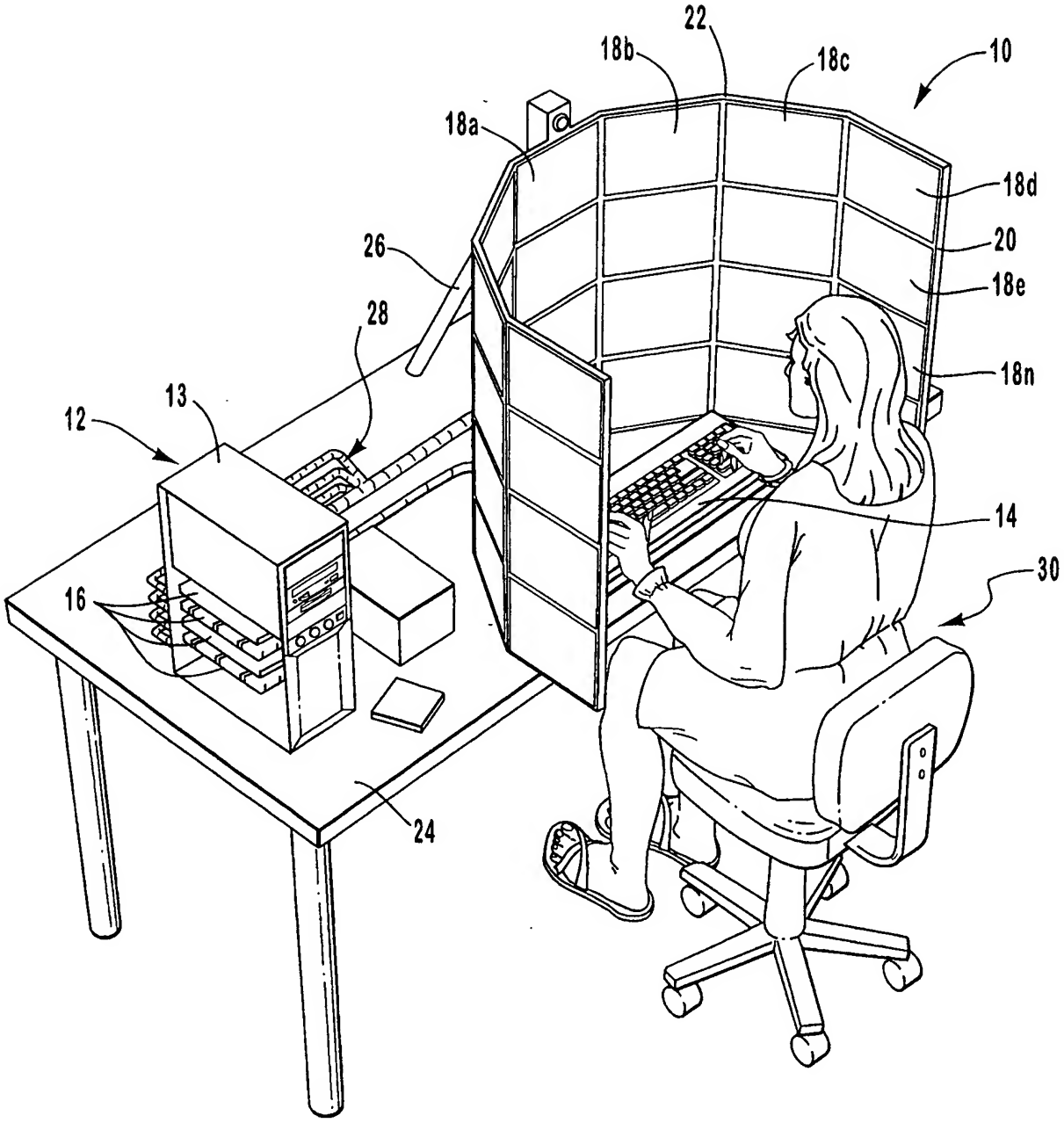


FIG. 1

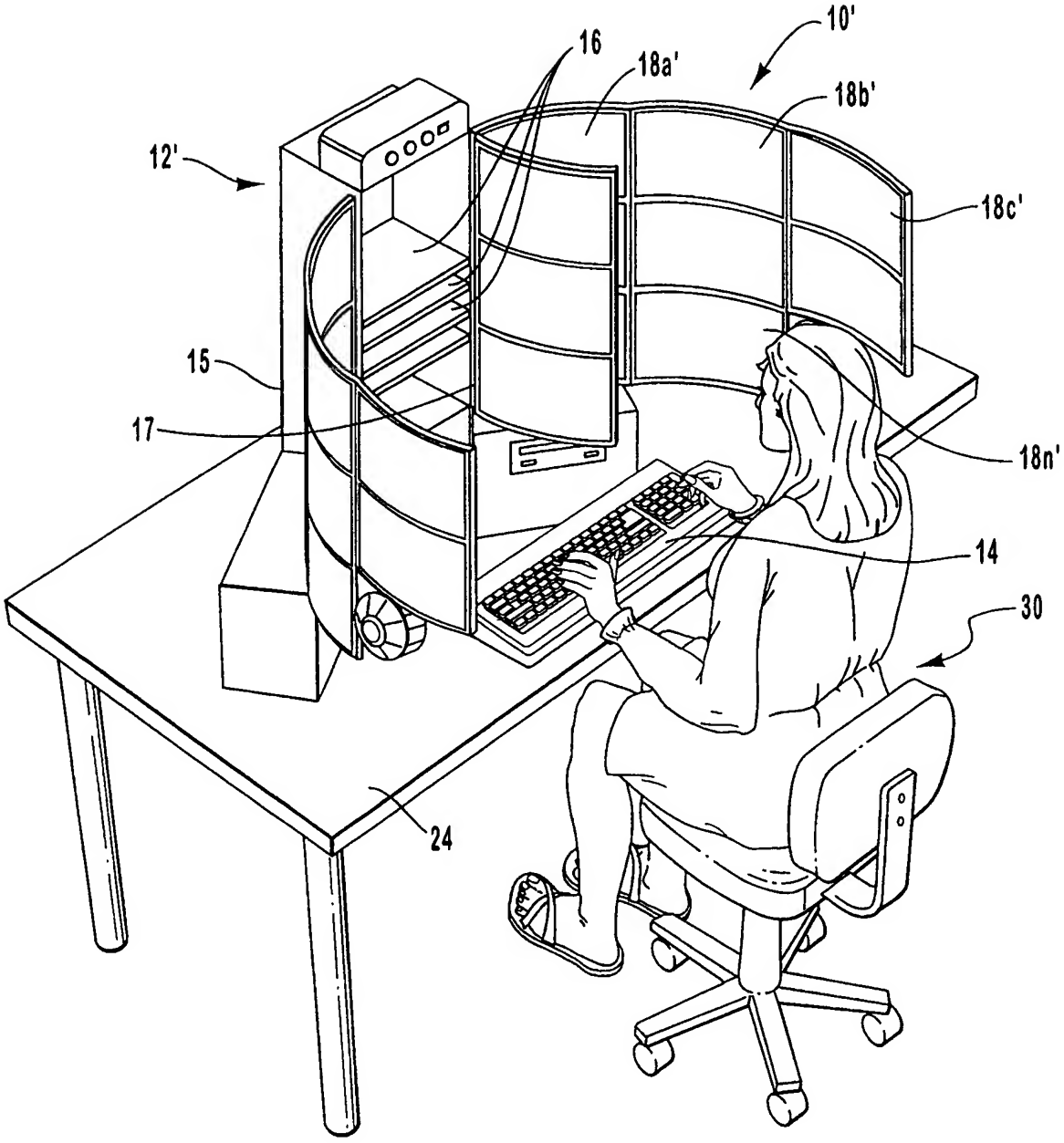


FIG. 2

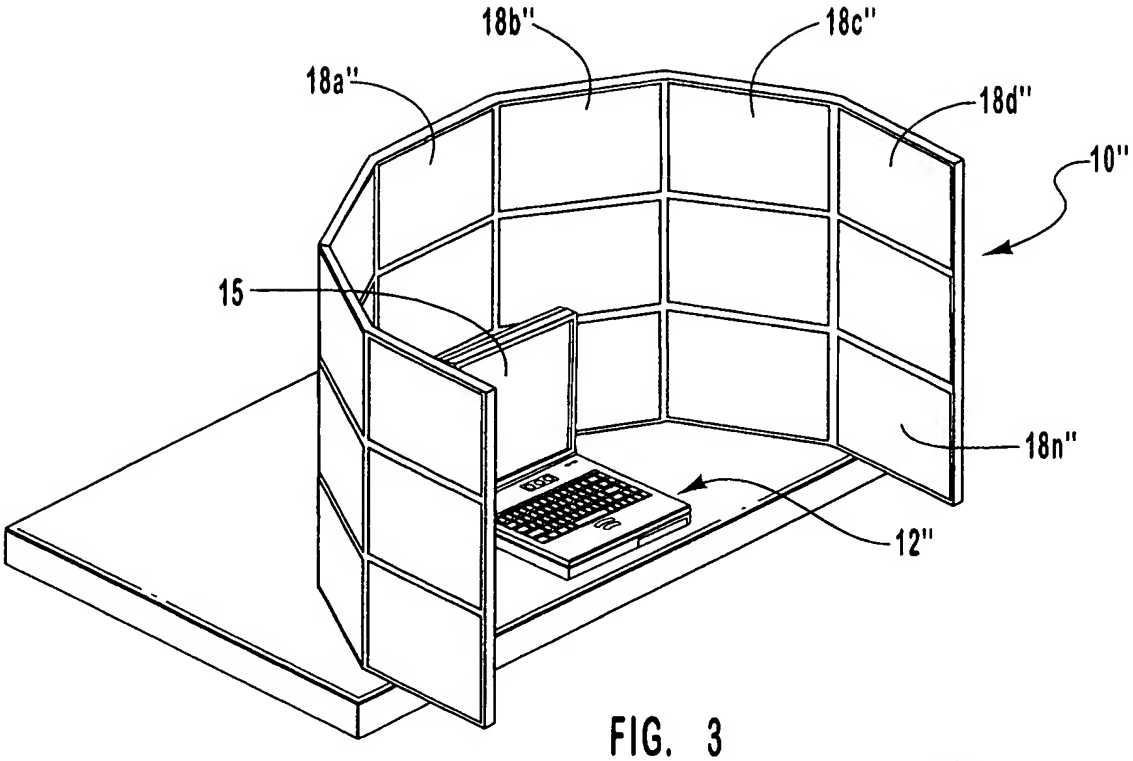


FIG. 3

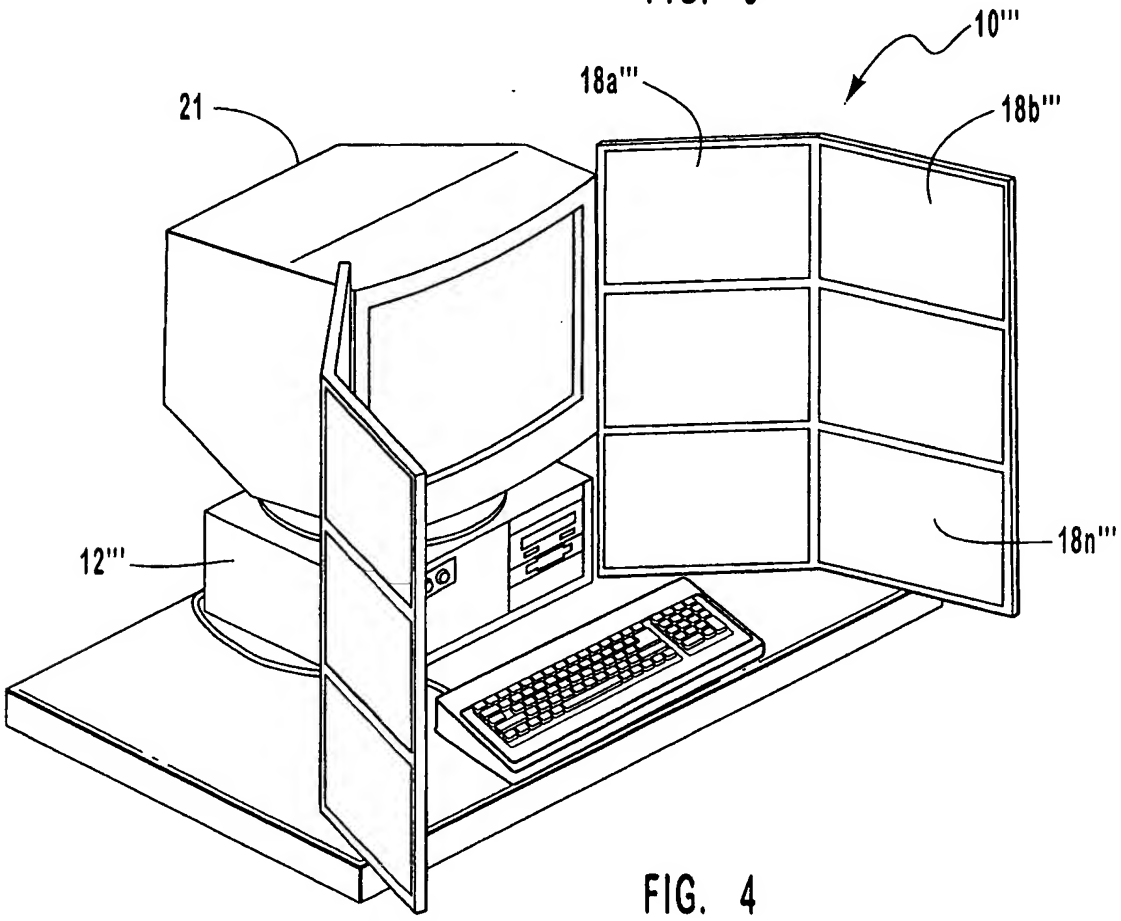


FIG. 4

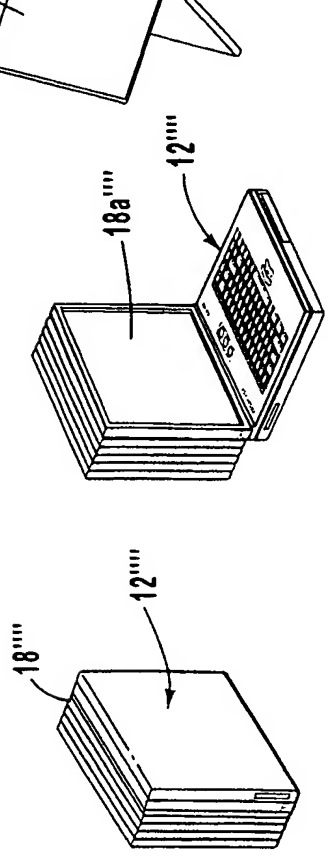


FIG. 5A

FIG. 5B

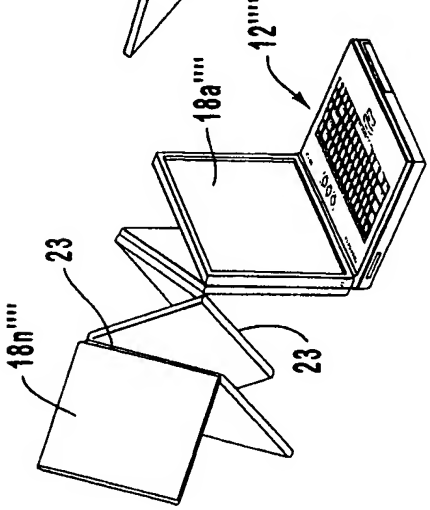


FIG. 5C

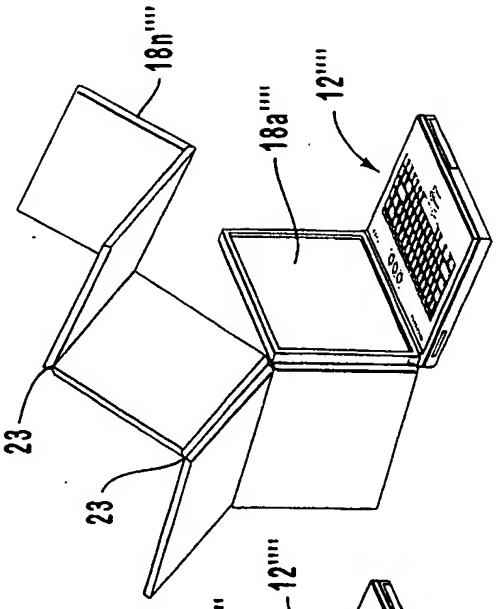


FIG. 5D

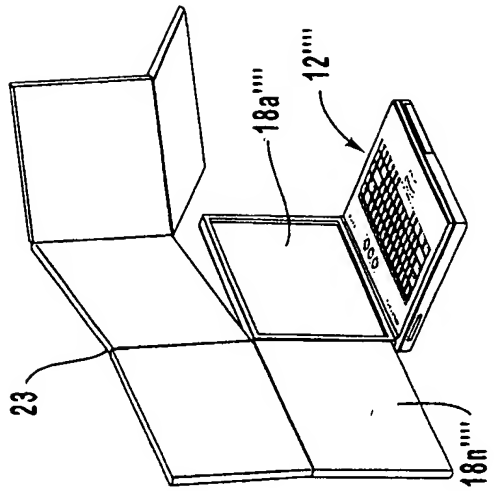


FIG. 5E

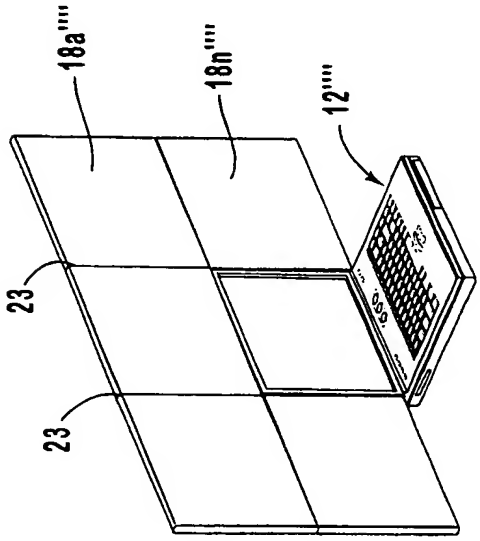


FIG. 5F

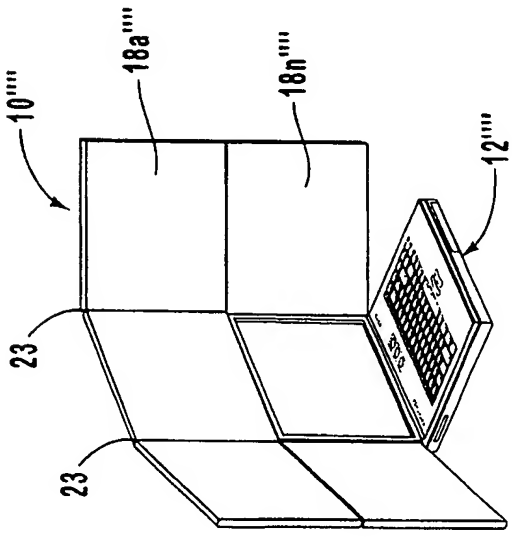


FIG. 5G

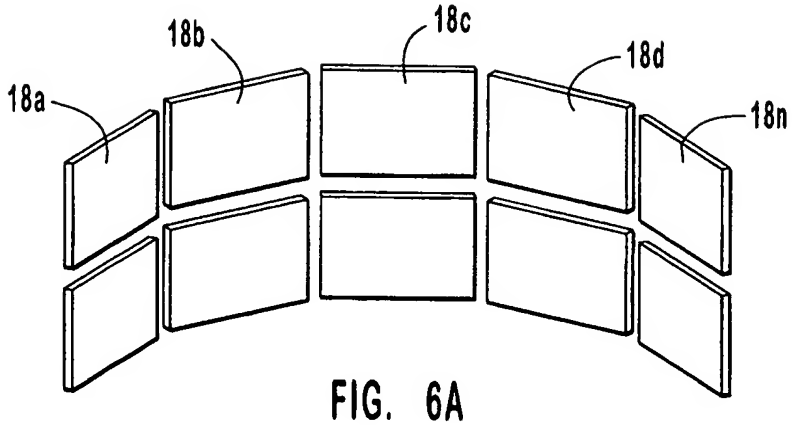


FIG. 6A

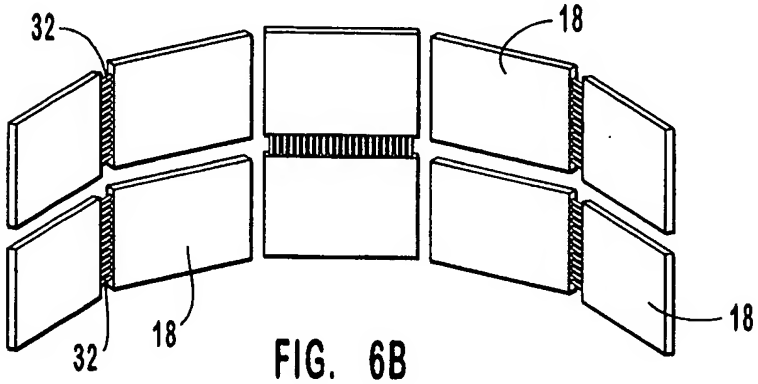


FIG. 6B

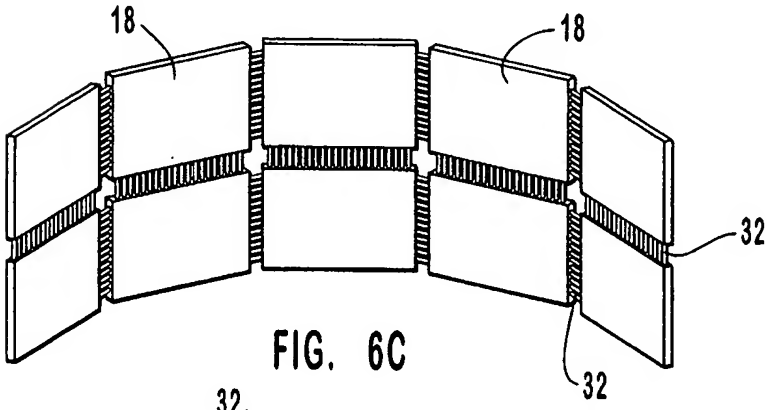


FIG. 6C

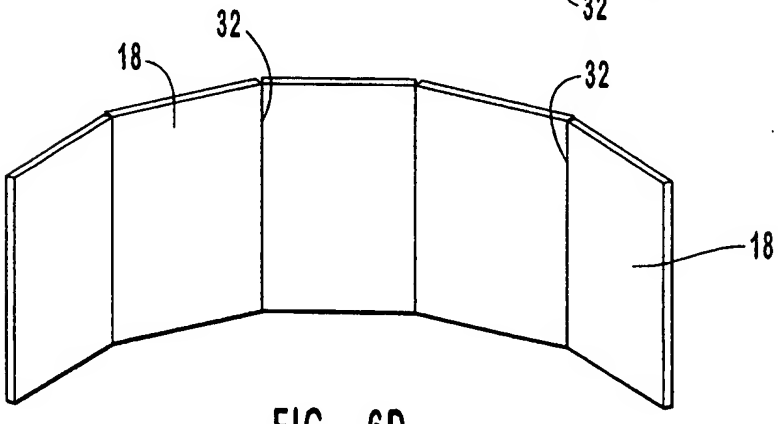


FIG. 6D

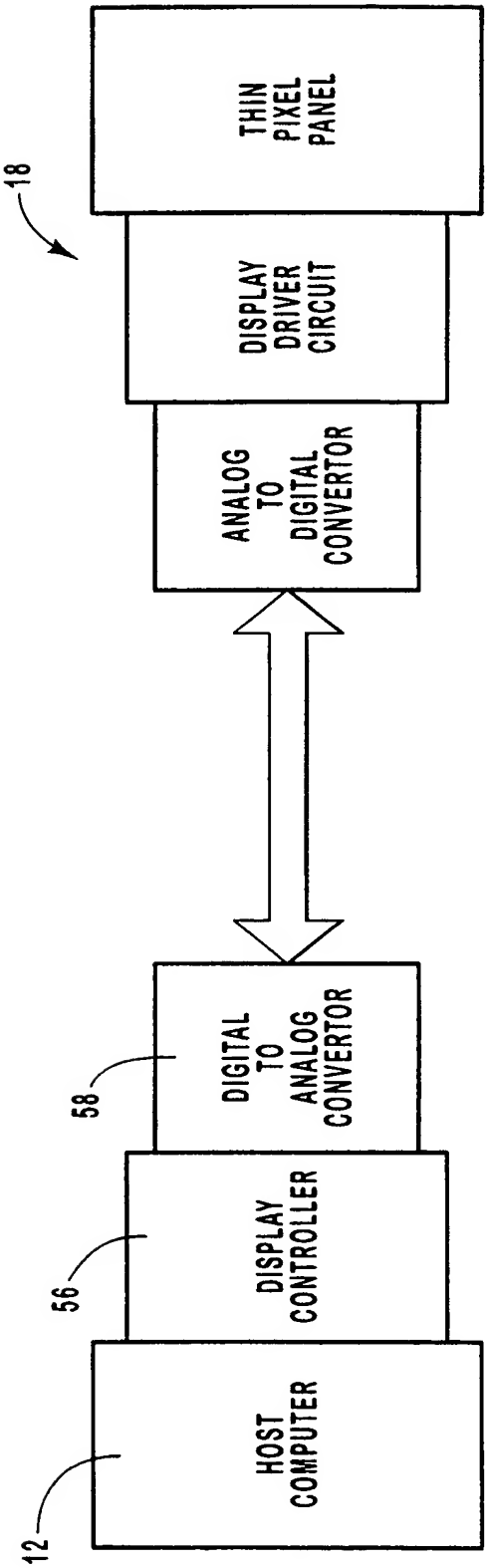

18
THIN PIXEL PANEL
DISPLAY DRIVER CIRCUIT
ANALOG TO DIGITAL CONVERTOR
58
DIGITAL TO ANALOG CONVERTOR
56
DISPLAY CONTROLLER
12
HOST COMPUTER

FIG. 7A

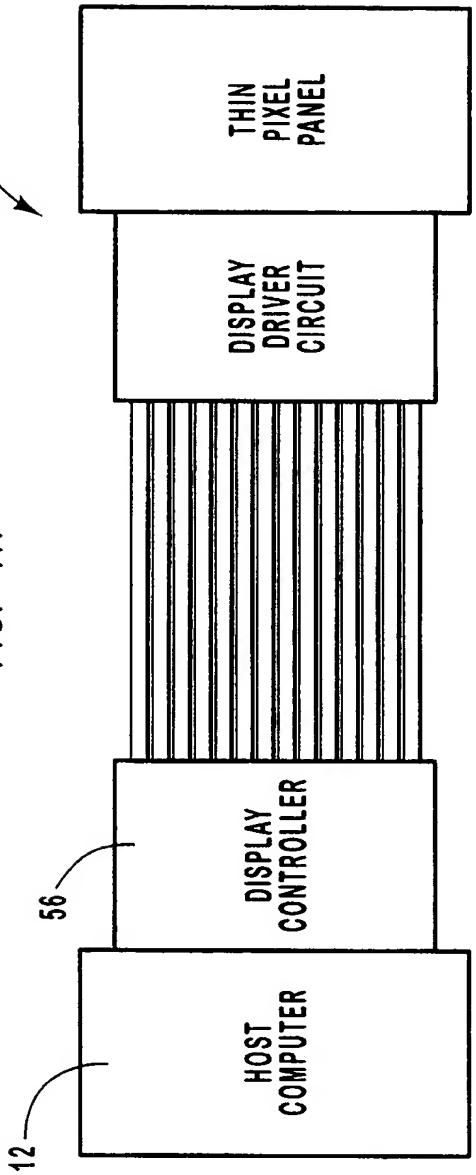

18
THIN PIXEL PANEL
DISPLAY DRIVER CIRCUIT
56
DISPLAY CONTROLLER
12
HOST COMPUTER

FIG. 7B

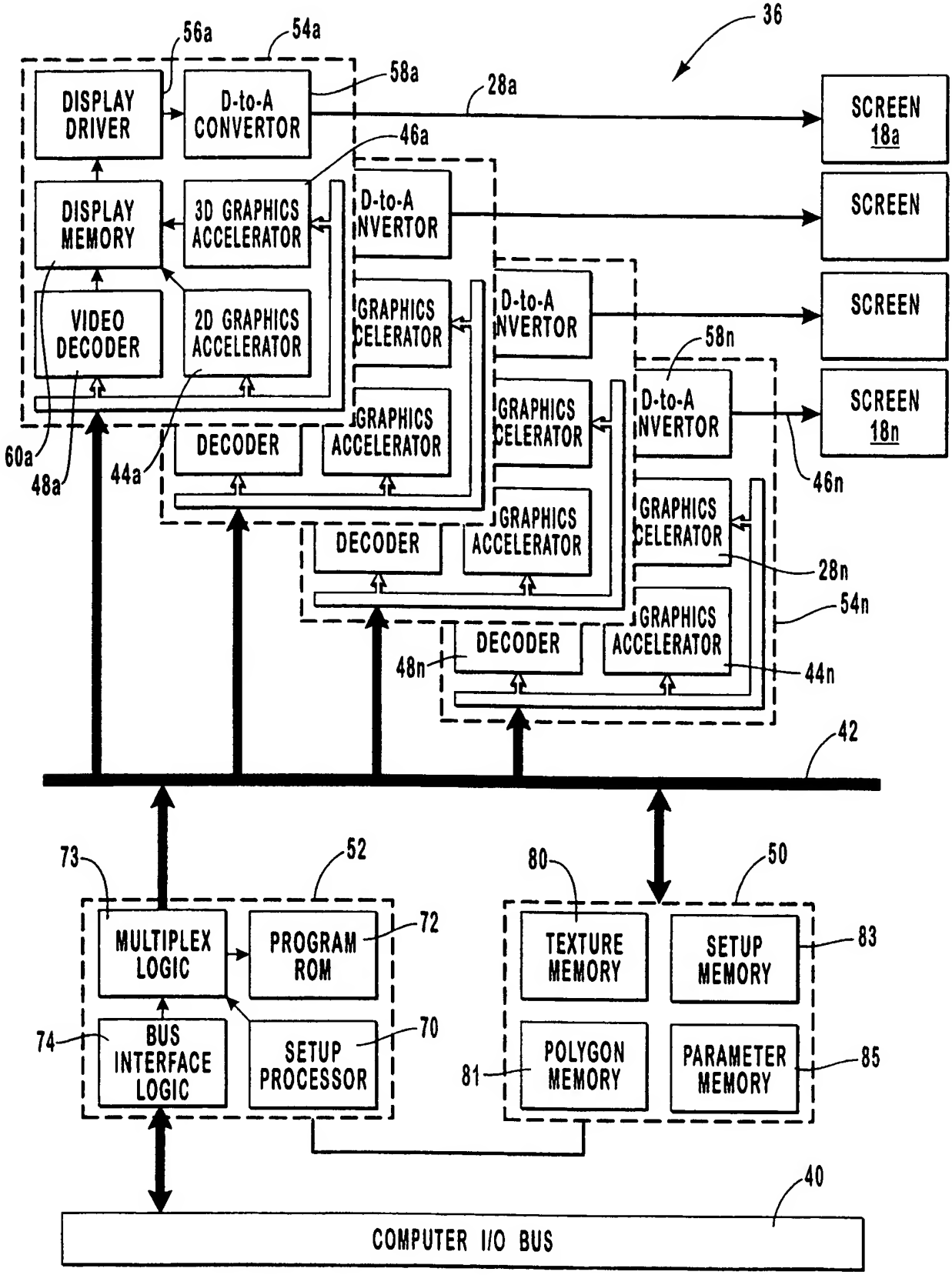

36
56a
54a
58a
28a
DISPLAY DRIVER
D-to-A CONVERTOR
SCREEN 18a
46a
DISPLAY MEMORY
3D GRAPHICS ACCELERATOR
D-to-A NVERTOR
SCREEN
VIDEO DECODER
2D GRAPHICS ACCELERATOR
GRAPHICS CELERATOR
D-to-A NVERTOR
SCREEN
58n
GRAPHICS ACCELERATOR
GRAPHICS CELERATOR
D-to-A NVERTOR
SCREEN 18n
60a
48a
44a
DECODER
46n
GRAPHICS ACCELERATOR
GRAPHICS CELERATOR
DECODER
28n
54n
GRAPHICS ACCELERATOR
48n
DECODER
44n
42
73
52
80
50
MULTIPLEX LOGIC
PROGRAM ROM
72
TEXTURE MEMORY
SETUP MEMORY
83
74
BUS INTERFACE LOGIC
SETUP PROCESSOR
70
81
POLYGON MEMORY
PARAMETER MEMORY
85
40
COMPUTER I/O BUS

FIG. 8

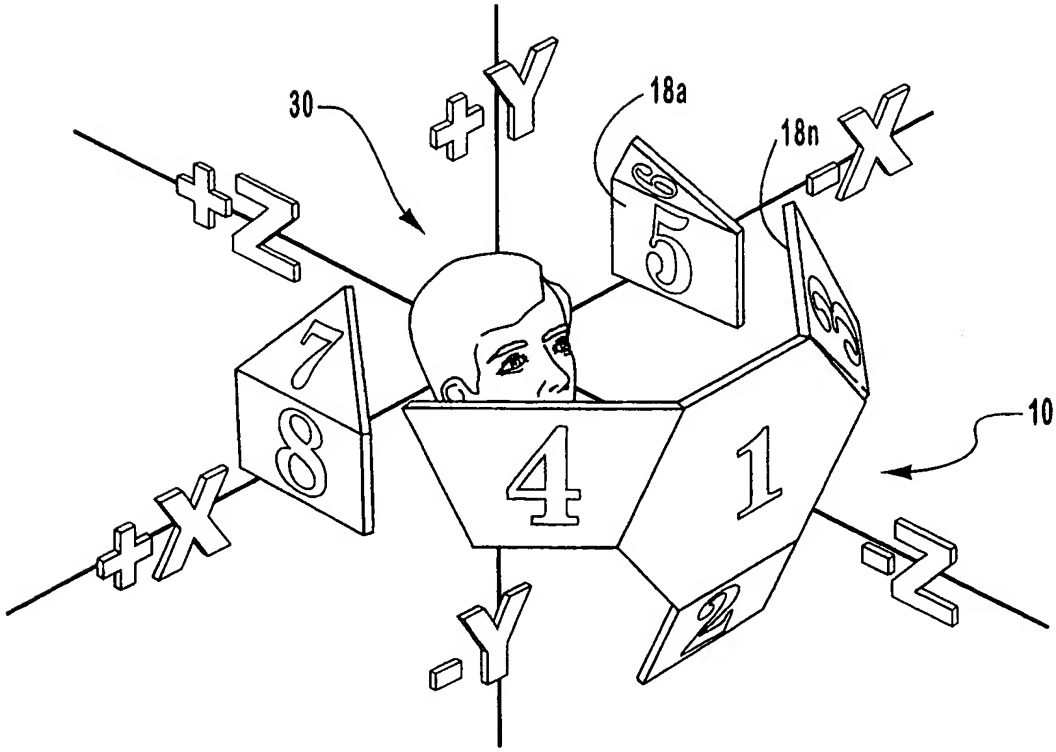


FIG. 9A

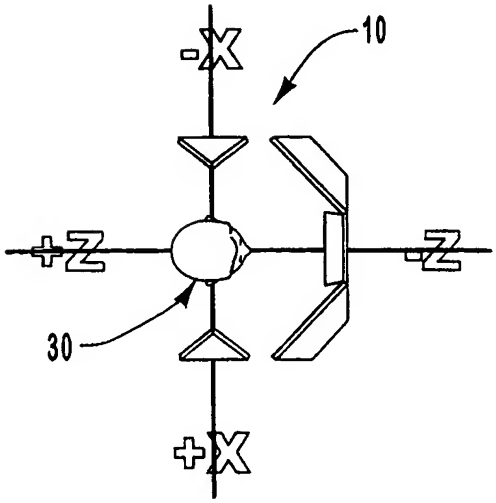


FIG. 9B

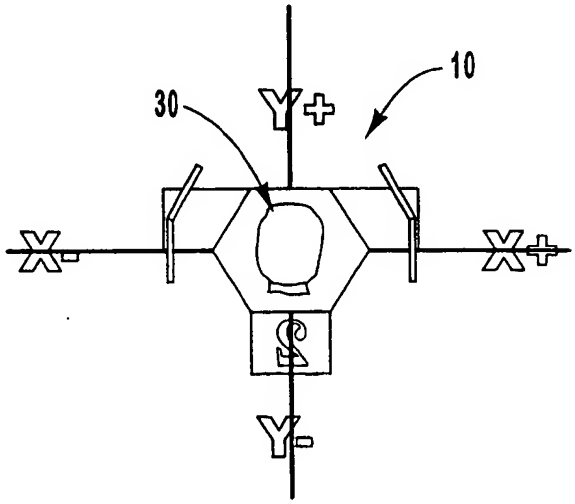


FIG. 9C

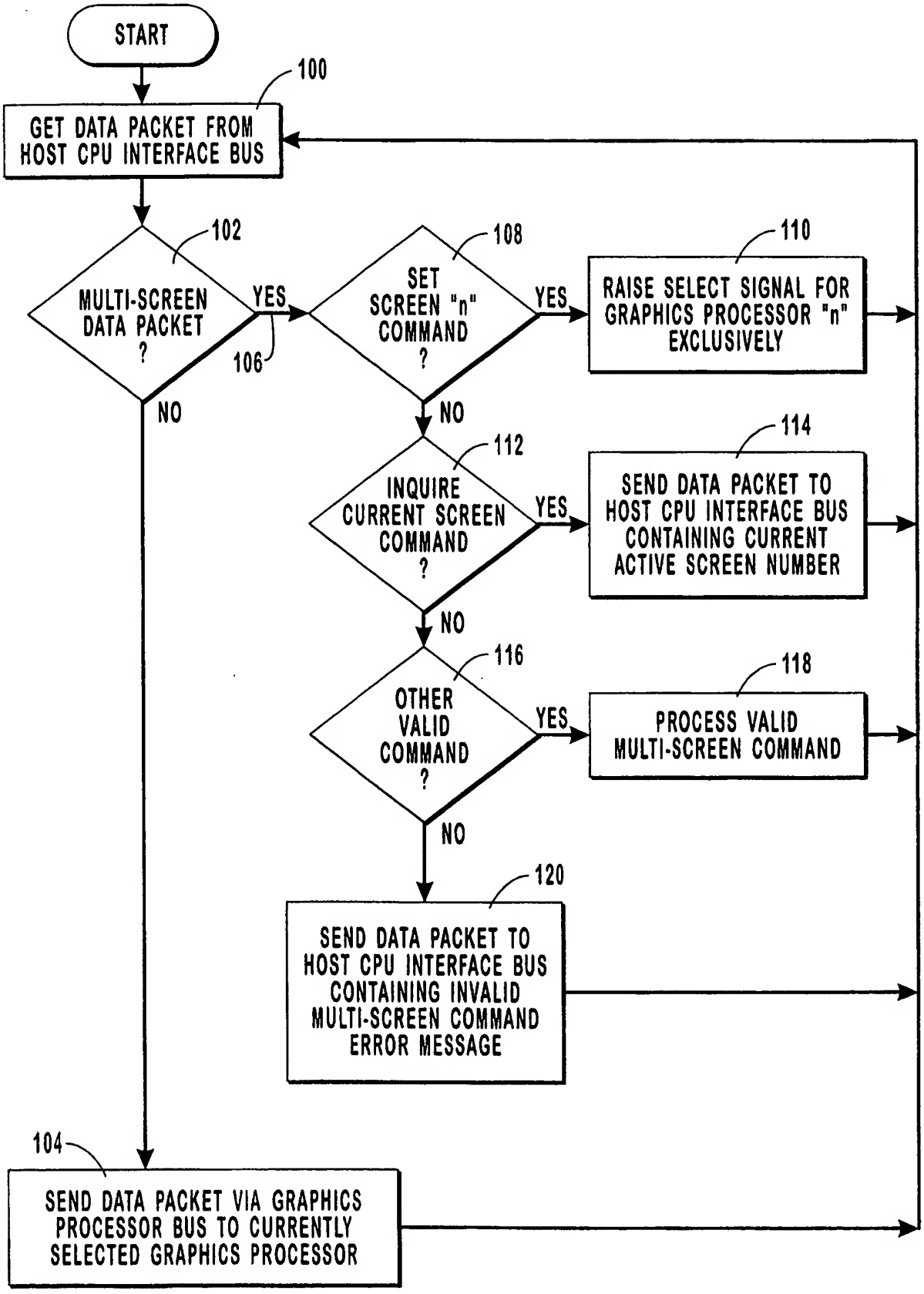

START
100
GET DATA PACKET FROM HOST CPU INTERFACE BUS
102
MULTI-SCREEN DATA PACKET ?
YES
106
NO
108
SET SCREEN "n" COMMAND ?
YES
110
RAISE SELECT SIGNAL FOR GRAPHICS PROCESSOR "n" EXCLUSIVELY
NO
112
INQUIRE CURRENT SCREEN COMMAND ?
YES
114
SEND DATA PACKET TO HOST CPU INTERFACE BUS CONTAINING CURRENT ACTIVE SCREEN NUMBER
NO
116
OTHER VALID COMMAND ?
YES
118
PROCESS VALID MULTI-SCREEN COMMAND
NO
120
SEND DATA PACKET TO HOST CPU INTERFACE BUS CONTAINING INVALID MULTI-SCREEN COMMAND ERROR MESSAGE
104
SEND DATA PACKET VIA GRAPHICS PROCESSOR BUS TO CURRENTLY SELECTED GRAPHICS PROCESSOR

FIG. 8

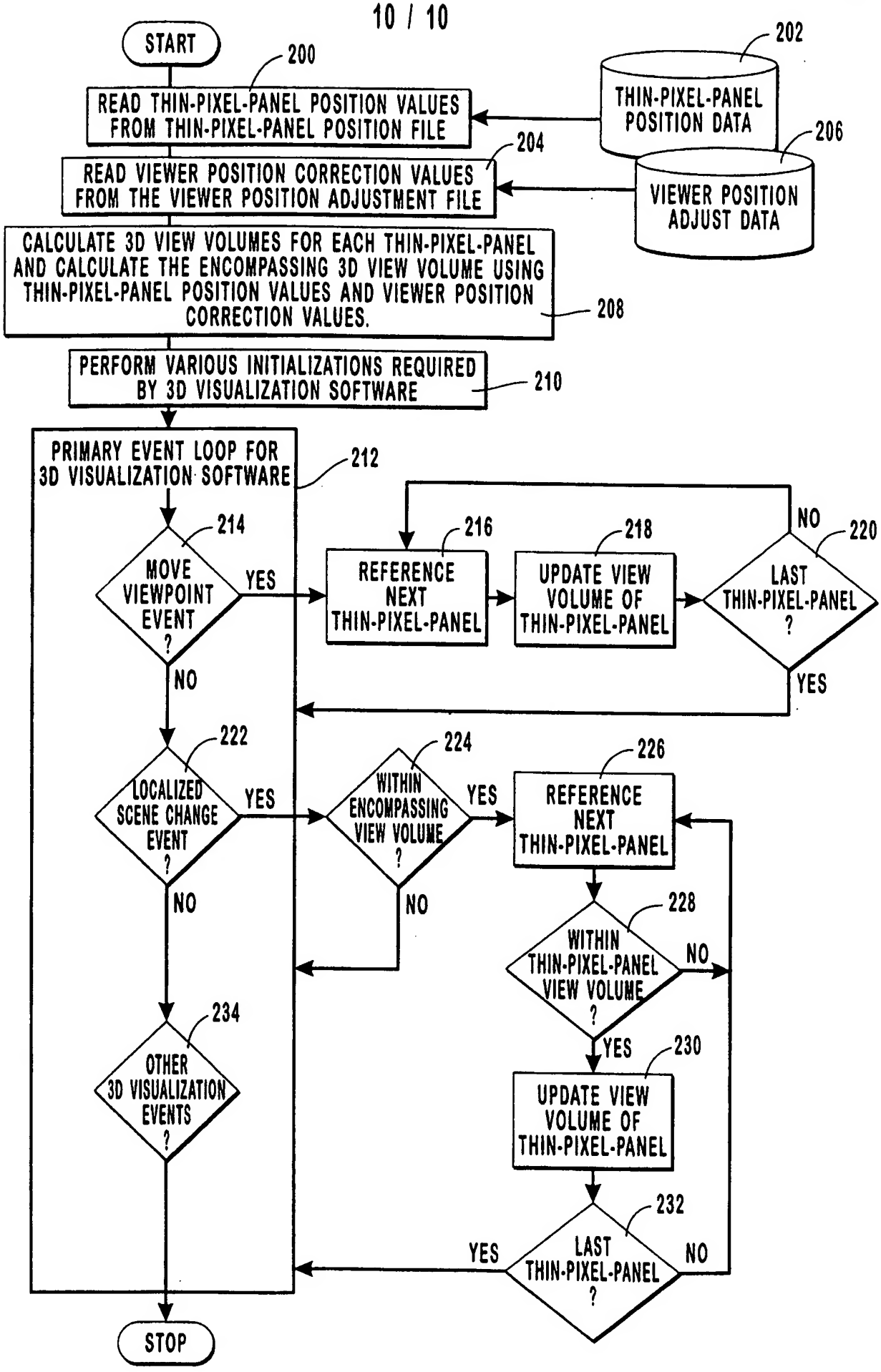

START
200
READ THIN-PIXEL-PANEL POSITION VALUES FROM THIN-PIXEL-PANEL POSITION FILE
202
THIN-PIXEL-PANEL POSITION DATA
204
READ VIEWER POSITION CORRECTION VALUES FROM THE VIEWER POSITION ADJUSTMENT FILE
206
VIEWER POSITION ADJUST DATA
CALCULATE 3D VIEW VOLUMES FOR EACH THIN-PIXEL-PANEL AND CALCULATE THE ENCOMPASSING 3D VIEW VOLUME USING THIN-PIXEL-PANEL POSITION VALUES AND VIEWER POSITION CORRECTION VALUES.
208
PERFORM VARIOUS INITIALIZATIONS REQUIRED BY 3D VISUALIZATION SOFTWARE
210
PRIMARY EVENT LOOP FOR 3D VISUALIZATION SOFTWARE
212
214
MOVE VIEWPOINT EVENT ?
YES
NO
216
REFERENCE NEXT THIN-PIXEL-PANEL
218
UPDATE VIEW VOLUME OF THIN-PIXEL-PANEL
220
LAST THIN-PIXEL-PANEL ?
NO
YES
222
LOCALIZED SCENE CHANGE EVENT ?
YES
NO
224
WITHIN ENCOMPASSING VIEW VOLUME ?
YES
NO
226
REFERENCE NEXT THIN-PIXEL-PANEL
228
WITHIN THIN-PIXEL-PANEL VIEW VOLUME ?
NO
YES
230
UPDATE VIEW VOLUME OF THIN-PIXEL-PANEL
232
LAST THIN-PIXEL-PANEL ?
YES
NO
234
OTHER 3D VISUALIZATION EVENTS ?
STOP

FIG. 11